Fall 12-15-2018

Accurate Vehicle Detection Using Multi-Camera Data Fusion and Machine Learning

Hao Wu
15180861737@163.com

ACCURATE VEHICLE DETECTION USING MULTI-CAMERA DATA FUSION AND MACHINE LEARNING

Approved by:

_____
Prof. Dinesh Rajan
Professor of Electrical Engineering

_____
Prof. Brett Story
Assistant Professor of Civil Engineering

_____
Prof. Carlos Davila
Associate Professor of Electrical Engineering

ACCURATE VEHICLE DETECTION USING MULTI-CAMERA DATA FUSION AND MACHINE LEARNING

A Thesis Presented to the Graduate Faculty of

Lyle School of Engineering

Southern Methodist University

in

Partial Fulfillment of the Requirements

for the degree of

Master of Science in Electrical Engineering

with a

Major in Electrical Engineering

by

Hao Wu

B.E., Electrical Engineering, Xidian University

December 15, 2018

ACKNOWLEDGMENTS

This work could not have been accomplished by the wisdom of my wonderful advisor, Prof. Dinesh Rajan, along with his many colleagues in the Department of Electrical Engineering here at SMU. I'm also forever grateful to my family for being so patient with me and my friends for reminding me to take breaks and go be great instead of just reading about greatness all of the time.

Hao, Wu B.S., Electrical Engineering, Xidian University, Xi'an 2015

Accurate Vehicle Detection Using Multi-camera
Data Fusion and Machine Learning

Advisor: Associate Professor Dinesh Rajan

Master of Science in Electrical Engineering conferred: December, 15, 2018

Thesis completed: November, 26, 2018

Computer-vision methods have recently been extensively used in intelligent transportation systems for vehicle detection. However, the detection of severely occluded or partially observed vehicles due to the limited camera fields of view remains a significant challenge.

This paper presents a multi-camera vehicle detection system that significantly improves the detection performance under occlusion conditions. The key elements of the proposed method include a novel multi-view region proposal network that localizes the candidate vehicles on the ground plane. We also infer the vehicle occupancies by leveraging multi-view cross-camera context. Experiments are conducted on a dataset captured from a roadway in Richardson, TX, USA, and the proposed system attains 0.7849 Average Precision (AP) and 0.7089 Multi Object Detection Precision (MODP). The proposed system advances the single-view region proposal approaches by approximately 31.2% for AP and 8.6% for MODP.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. Introduction

1.1 Background

Vision-based vehicle detection methods have recently received significant attention in intelligent transportation systems (ITSs). Reliable vehicle detection is a fundamental component of traffic surveillance with increased safety and mobility implications [1]. A comprehensive review of vehicle detection system is given in [2]. In existing researches on vehicle detection, locating multiple vehicles in crowded traffic scenes is a challenging task due to the limited field-of-view of the camera. Specifically, the open research problem is to detect occluded or partially-observed vehicle in the 2D view that is obtained from a single camera view point.

One way to overcome the challenge of detecting partially-occluded vehicles is to detect the candidate vehicles using their multiple semantic sub-parts [10], [11], [12]. Although these methods adapt to situations with partial occlusions, they fail when vehicles are severely occluded in traffic dynamics [8], [9], [14]. Another feasible way to overcome the occlusion challenge is to use a multi-camera system and fuse the information from each independent camera stream [13], [17]. Such methods are based on a hypothesis that objects occluded in some views may not be occluded in other views. Recent algorithms on multi-camera object detection mainly focus on pedestrian detection. These algorithms infer the pedestrian locations on the ground plane by extracting monocular features and estimating the ground-plane occupancy vector. In order to estimate the ground-plane occupancy vector, some of the multi-camera object detection systems extract binary foreground mask as the feature, which is not robust in severely-occluded traffic scenes [15], [16],

[18]. Some other algorithms use features that are generated by a deep Convolutional Neural Network (CNN) [19], [20]. The existing approaches fuse the extracted features to infer the occupancy vector. The location of a pedestrian is represented using a single cell (with predefined shape and size) in the ground-plane[15], [16], [18], [19], [20]. The fixed-size cells are appropriate for detecting pedestrians due to the similarity of the footprint of various pedestrians on ground plane. However, using fixed cells to detect vehicles that have large variations in shape and size, e.g. truck vs. sedan on ground plane is not appropriate.

Therefore, to address the aforementioned issues, this thesis develops: 1) a Multi-View Region Proposal Network (MVRPN) to estimate the ground-plane occupancy vector by leveraging multiple side views simultaneously, and 2) a fine-tuned pre-trained deep CNN to remove false positive object predictions that are generated by the trained MVRPN. In the proposed system, the MVRPN is trained by using given ground-plane information, which is captured from a top-view camera. Instead of using a single cell with predefined size, the location of objects on the ground plane are represented by cell blocks with adaptive size. Therefore, the proposed system can be applied to vehicles with large variations in size. We also use AlexNet as the basis for transfer learning to derive the CNN required in this work. Our experimental results demonstrate that using 3 cameras with different vantage points provides an improvement in the accuracy of detecting vehicles over a system that uses just 1 camera. We also quantify the improvement obtained when only images from 2 of the 3 cameras are used in the detection process. As expected there is a further improvement in performance when going from 2 to 3 cameras.

1.2 Related Work

In single-view object detection methodology, Wang and Fang [12] propose a part-based vehicle detection system that uses probabilistic inference to address the issues of partial observation and varying viewpoints. This system consists of two parts, viewpoint-discriminative part-based geometric appearance models (VDPAM) and viewpoint-discriminative part based geometric model (VDPGM). The training process of VDPAM and VDPGM are conducted in an off-line manner. In online detection, the major part of vehicle is first detected by utilizing VDPAM, and then a probabilistic representation that describe the configuration of vehicle parts as well as their spatial relations is then conducted by exploiting VDPGM. Such an approach can achieve a promising result to detect partial occlusion vehicle. However, this method will fail when the major part of vehicle is occluded so that VDPAM can not detect them.

The work flow of multi-view objects detection task is to first extract features from surveillance video and then predicting the 3D location of the vehicle depending on those features. To do so, previous works usually integrate object information from each view to the reference plane by utilizing camera calibration information. The ground plane, e.g. the plane with $z = 0$, is often selected as the reference plane. In this section, we will briefly review previous related works that integrates information to the ground plane.

In an early study of multi-view pedestrian detection, Kim and Davis [15] focused on refining the single-view pedestrian detection results with multi-view homography. They projected the detection results from all views to the same ground plane to find their intersection points, which were treated as the pedestrians' locations. Since it is often difficult to accurately detect pedestrians from each single view, the method applied in [15] only approximately detect the foreground regions from each view, while complicated analysis is conducted on the ground plane to locate

pedestrians in these foreground regions. Generally speaking, such an approach is very efficient and outperform many single-view objects detection approaches in the scenarios when the distribution of pedestrian is sparse. However, it may fail when the scenes become extremely crowd since many false positive predictions may arise in a crowded scene, which should be further distinguished from real objects.

To solve this problem, the approach in [18] utilizes a multi-view Bayesian network to remove those false positive predictions. A set of preliminary detection results using the existing multi-view pedestrian detection methods are first obtained. Such results can be represented as the pedestrian candidates in all view and the corresponding locations on the ground plane. After that, a Bayesian network is utilized to model the occlusion relationship among all candidates in each camera view, and then multiple Bayesian networks can be further combined to infer the false positive prediction. However, both methodologies in [15] and [18] extract binary foreground mask from each single view as the feature, which is not robust in severely-occluded traffic scenes.

Beyond these approaches, many recent studies utilize feature maps generate by the convolution layer of a deep CNN to infer the candidate pedestrian locations on the ground plane. In [19], Baqué and Fleuret use such feature map as the input to train a Conditional Random Field (CRF) which can explicitly model the occlusions between each pedestrian on the ground plane. In [20], Chavdarova also utilizes the feature map generated by the deep CNN. However, instead of using CRF, a Multi-Layer Perceptron (MLP) prediction network is utilized to infer the location of candidate pedestrians on the ground plane. However, in [19], the occupancy vector on the ground plane is obtained from each side view independently; and in [20], the estimation of the multi-view joint occupancy takes higher computations when projecting every ground-plane cell back to each side view.

The remainder of this Thesis is organized as follows. Chapter 2 presents the descriptions of the proposed multi- view vehicle detection system. Experiments and results are provided in Chapter 3, followed by conclusions in Chapter 4.

II. Proposed System Description

The core objective of the proposed system is to localize the vehicles on the ground plane by fusing synchronized frames from a multi-camera network. An overview of the proposed system is shown in Fig.1, and the frequently used notations are given in Table 1. A MVRPN is introduced to deduce the candidate vehicle Region of Interests (ROIs) on the ground plane from side-view images. A multi-view ROI inference is then used to obtain the probability of the deduced ROIs being a vehicle.

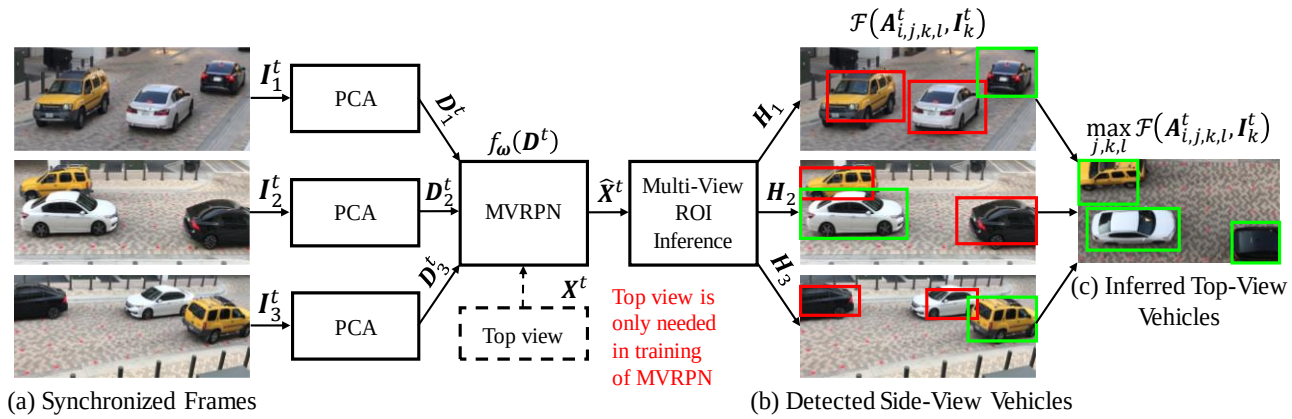


Fig. 1. The overview of the multi-camera vehicle detection system. The original synchronized frames from 3 side cameras are shown in (a). The detected vehicles on side views and the inferred vehicles on top view are shown in (b) and (c). The top-view vehicles are inferred by the corresponding detections with the maximum probabilities, which are the green boxes in (b).

Consider a camera network composed of C side-view cameras and 1 top-view camera, where each camera can have a different resolution. The top-view camera is used to capture the ground-truth information from the ground plane without any occlusion to train the MVRPN and also to quantify the performance of the proposed algorithm; a top-view camera is not necessary

for field implementation of a trained system. From the top-view camera, the t^{th} ground-plane image, I^t_{top} which is of size $N_G \times M_G \times 3$ is captured. A foreground binary mask is then obtained by binary pixel-wise labeling of the ground-plane frame into the vehicle and non-vehicle class. The binary mask of the ground-plane frame is subsampled into a grid of size $\frac{N_G}{m} \times \frac{M_G}{m}$, where m is a hyper parameter to adjust the size of grid of cells while making its aspect ratio to be identical with the ground-plane frame. In our experiments, we set $m = 20$, and the total number of cells is $N = \frac{N_G}{m} \times \frac{M_G}{m}$. We denote the grid of cells as a 2-D binary matrix $\boldsymbol{G}^t$, where the matrix element with value equal to 1 represents that the corresponding cell is occupied by a vehicle. By concatenating columns of $\boldsymbol{G}^t$, the $N \times 1$ ground-truth Boolean occupancy vector is obtained. We denote occupancy vector as $\boldsymbol{X}^t$, where $\boldsymbol{X}^t = \{X_1^t, X_2^t, \dots X_N^t\}^T$. Note that the superscript t of those notations is the index into the set of captured frames. The t^{th} RGB frame captured from side-view camera k is denoted as $\boldsymbol{I}_k^t$ with size equal to $N_k \times M_k \times 3$, where $k \in \{1,2, \dots, C\}$. The large dimension of the input $\boldsymbol{I}_k^t$ increases the unknown training parameters and makes the MVRPN computationally hard to converge [21]. Hence, Principle Component Analysis (PCA) [22] is used to reduce the dimension of frames that are captured from each side view camera.

Table 1. Frequently Used Notations

	Description
$\boldsymbol{I}_k^t$	The t^{th} RGB frame from side view k.
$\boldsymbol{G}^t$	The 2-D binary ground-plane grid of cells.
$\boldsymbol{X}^t$	The 1-D ground-truth Boolean occupancy vector.
$\boldsymbol{D}^t$	The dimension-reduced input vector.
$\widehat{\boldsymbol{X}}^t$	The estimated ground-plane occupancy vector.
$\boldsymbol{R}_i^t$	The i^{th} MER on the ground plane.
$\boldsymbol{H}_k$	The homography between k^{th} side view and ground plane.
$\boldsymbol{C}_{i,j}^t$	The j^{th} foreground cells in $\boldsymbol{R}_i^t$ on the ground plane.
$\boldsymbol{P}_{i,j,k}^t$	The projection of top-left corner $\boldsymbol{C}_{i,j}^t$ in $\boldsymbol{R}_i^t$ at side view k.
$\boldsymbol{A}_{i,j,k,l}^t$	The l^{th} bounding box with bottom edge centered at $\boldsymbol{P}_{i,j,k}^t$.
$f_{\boldsymbol{\omega}}(\cdot)$	The function that represents the MVRPN.
$\mathcal{F}(\cdot)$	The fine-tuned pre-trained deep CNN classification.

*Note: $\boldsymbol{R}_i^t$, $\boldsymbol{C}_{i,j}^t$ and $\boldsymbol{A}_{i,j,k,l}^t$ are 4-element vectors that represent the selected rectangular bounding boxes with the form $[x_{\min}\ y_{\min}\ \omega\ h]$.

2.1 Principle Component Analysis

The PCA algorithm is a popular technique which can be leveraged to quantitatively analyze the correlation between different variables in the data. The main goal of PCA is to find the directions of maximum variance of data. Such directions can be represented by a set of orthogonal vectors called principal components. In this research, PCA algorithm is applied to reduce the dimension of each frames captured from the various cameras.

For the camera k in the camera network, $\boldsymbol{I}_k^t$ is transformed into a grayscale image and a $1 \times N_k M_k$ row vector, $\boldsymbol{v}_k^t$ where $k \in \{1,2,\dots,C\}$, is generated by stacking each row in the grayscale image together. After all the data is captured, let $\boldsymbol{V}_k = (\boldsymbol{v}_k^1, \boldsymbol{v}_k^2, \dots, \boldsymbol{v}_k^B)^T$ denotes the

data matrix with size equal to $B \times N_k M_k$ which composed of all data captured from camera k, and B is the total number of frames captured from camera k. The principle components of the data are essentially the eigenvectors of the covariance matrix of $\boldsymbol{V}_k$. Let $\sum_k$ denotes the covariance matrix of $\boldsymbol{V}_k$ with size equals to $N_k M_k \times N_k M_k$, and can be computed as:

$$\sum_k = \frac{1}{B-1}\left\{\left(\boldsymbol{V}_k - \overline{\boldsymbol{V}}_k\right)^T \left(\boldsymbol{V}_k - \overline{\boldsymbol{V}}_k\right)\right\}, \tag{2.1}$$

where $\overline{\boldsymbol{V}}_k$ is the column-wise mean vector of $\boldsymbol{V}_k$ whose size equals to $1 \times N_k M_k$, where

$$\overline{\boldsymbol{V}}_k = \frac{1}{B}\sum_{t=1}^{B} \boldsymbol{v}_k^t \tag{2.2}$$

After the covariance matrix $\sum_k$ is obtained, an eigen-decomposition algorithm is leveraged to calculate the eigenvectors and eigenvalues of $\sum_k$. Let $\boldsymbol{e}_i$ denotes the i^{th} eigenvector of $\sum_k$ whose size is $N_k M_k \times 1$ and $\curlywedge_i$ denotes the corresponding eigenvalue, where $i \in \{1,2,\dots,N_k M_k\}$. Also we let $\boldsymbol{E}_k = (\boldsymbol{e}_1, \boldsymbol{e}_2, \dots, \boldsymbol{e}_{N_k M_k})$ denotes a $N_k M_k \times N_k M_k$ matrix, each column of the matrix $\boldsymbol{E}$ is the eigenvector of $\sum_k$, and $\boldsymbol{L_k} = \mathrm{diag}(\curlywedge_i)$ denotes the $N_k M_k \times N_k M_k$ diagonal matrix whose entries on the main diagonal are the eigenvalues corresponding to $\boldsymbol{e}_i$, and all entries on the diagonal are organized in decreasing order. After eigen-decomposition procedure, we have

$$\sum_k \boldsymbol{E}_k = \boldsymbol{E}_k \boldsymbol{L_k} \tag{2.3}$$

In order to reduce the dimension of all frames captured from camera k, the accumulated variance of data on each principle components have to be measured by

$$\beta_k^p = \frac{\sum_{i=1}^{p} \curlywedge_i}{\sum_{i=1}^{N_k M_k} \curlywedge_i} \tag{2.4}$$

Where the denominator of equation (2.4) is the total variance of data on each principle component and the numerator is the accumulated variance of data from the first principle component to the p^{th} principle component. The visualization of accumulated variance of data captured by one camera is shown in Fig.2.

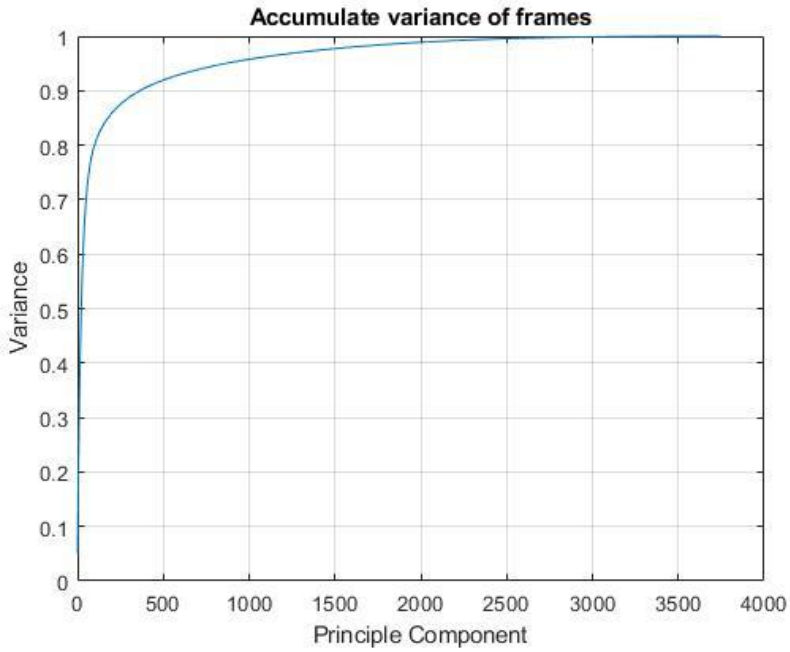


Fig.2 Visualization of accumulated variance on each principle component

Due to the highly correlation between neighbor pixel, after projecting the grayscale frame to the principle components, the first 500 principle components still retain more than 90% information of raw grayscale frames. Hence in this research, Principle Component Analysis (PCA) [22] is used to generate $\boldsymbol{D}_k^t$, a n_k dimensional column vector from $\boldsymbol{I}_k^t$, where $n_k \ll N_k \cdot M_k \cdot 3$, and we set $n_k = 500$.

2.2 Multi-View Region Proposal Network (MVRPN)

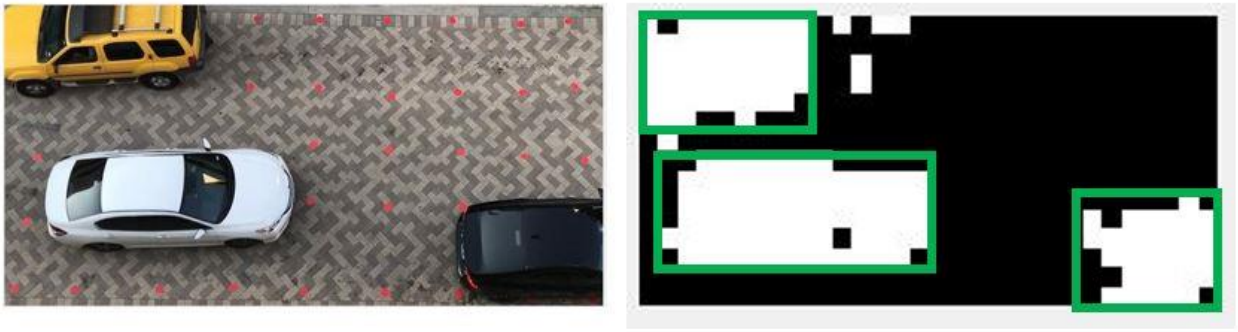
After the PCA procedure, the input column vector $\boldsymbol{D}^t = \{\boldsymbol{D}_1^t, \boldsymbol{D}_2^t, \dots, \boldsymbol{D}_c^t\}^T$ of the MVRPN is obtained, where $\boldsymbol{D}^t$ is composed of C dimension-reduced vector of frames captured from different side-view cameras at the same time. Given $\boldsymbol{D}^t$, a Multi-Layer Perceptron (MLP) architecture, MVRPN, is utilized to estimate the ground-plane occupancy vector, $\widehat{\boldsymbol{X}}^t = \{\hat{X}_1^t, \hat{X}_2^t, \dots, \hat{X}_N^t\}^T$. In the proposed system, we assume that the number of cells occupied by vehicles on the ground plane is lesser than those corresponding to the background. Therefore, due to the imbalanced vehicle instances, the training process of MVRPN suffers from the bias problem [23]. To alleviate this issue, the loss function $\boldsymbol{\mathcal{L}}$ in training the MVRPN is set as:

$$\boldsymbol{\mathcal{L}}_{\boldsymbol{\omega}}\left(\boldsymbol{X}^t, \widehat{\boldsymbol{X}}^t\right) = \begin{cases} \dfrac{\alpha}{2N}\displaystyle\sum_{i=1}^{N}\left[\hat{X}_i^t - X_i^t\right]^2, if\ X_i^t = 1 \\ \dfrac{1}{2N}\displaystyle\sum_{i=1}^{N}\left[\hat{X}_i^t - X_i^t\right]^2, otherwise \end{cases} \tag{2.5}$$

where

$$\widehat{\boldsymbol{X}}^t = f_{\boldsymbol{\omega}}(\boldsymbol{D}^t) = \{\hat{X}_1^t, \hat{X}_2^t, \dots, \hat{X}_N^t\}^T \tag{2.6}$$

The loss function $\boldsymbol{\mathcal{L}}_{\boldsymbol{\omega}}\left(\boldsymbol{X}^t, \widehat{\boldsymbol{X}}^t\right)$ is the weighted Mean Squared Error (WMSE) between the estimated $\widehat{\boldsymbol{X}}^t$ and the ground truth $\boldsymbol{X}^t$. The operation of the MVRPN is denoted in functional form as $f_{\boldsymbol{\omega}}(\boldsymbol{D}^t)$, where $\boldsymbol{\omega}$ are the MVRPN parameters to be learned, and $\widehat{\boldsymbol{X}}^t$ is the output of MVRPN. The penalization weight α adaptively applies more penalty to the computed WMSE when MVRPN classifies a foreground cell as background, i.e. in this study, $\alpha = 5$. The output of MVRPN and the corresponding frame captured from top-view camera is shown in Fig.3.



(a) Frame from top-view camera (b) Corresponding Output of MVRPN

Fig.3 Frame from top-view camera and the corresponding output from MVRPN, the green bounding box in Fig.3 (b) are Minimum Enclosing Rectangles (MERs) that represent the candidate ROIs.

After estimating the occupancy vector $\widehat{\boldsymbol{X}}^t$, a set of candidate ROIs, which are Minimum Enclosing Rectangles (MERs) to enclose foreground cells block, are generated. Examples of such MERs are shown in Fig.3. In this paper, the i^{th} MER of $\widehat{\boldsymbol{X}}^t$ is denoted as $\boldsymbol{R}_i^t$, where $i \in \{1, 2, \dots, M\}$, and M is the total number of MERs on the ground plane. The j^{th} foreground cells in i^{th} MER is denoted as $\boldsymbol{C}_{i,j}^t$, where $j \in \{1, 2, \dots, P\}$ and P is the number of foreground cells within $\boldsymbol{R}_i^t$. However, since some MERs are false positives (FPs), a multi-view ROI inference is leveraged to remove those FPs. For this purpose, a set of homography matrices are estimated by using RANSAC and Levenberg-Marquardt algorithms [24].

2.3 Computing the Homography between side view camera and ground plane

A 2D point (x, y) in an image can be represented as a 3D vector $\mathbf{x} = [x_1, x_2, x_3]^T$ where $x = \frac{x_1}{x_3}$ and $y = \frac{x_2}{x_3}$. This is called the homogeneous coordinate of a point and it lies on the projective plane. A homography is an invertible mapping of points and lines on the projective

plane. Hartley and Zisserman in [30] define the homography as a non-singular 3×3 matrix such that for any point in P^2 represented by vector $\mathbf{x}$ it is true that its mapped point equals $\boldsymbol{H}\mathbf{x}$, where $\boldsymbol{H}$ is the 3×3 homography matrix and P^2 denotes the projective plane in which $\mathbf{x}$ lies.

In this research, the ground plane frame that is captured from the top-view camera is set to be the reference image. The homography matrix that can map the corresponding points from ground plane to k^{th} side view is denoted as $\boldsymbol{H}_k$, where,

$$\boldsymbol{H}_k = \begin{bmatrix} h_k^1 & h_k^2 & h_k^3 \\ h_k^4 & h_k^5 & h_k^6 \\ h_k^7 & h_k^8 & h_k^9 \end{bmatrix} \tag{2.7}$$

and $k \in \{1,2,\dots,C\}$ and C is the number of side-view cameras. To calculate the homography, we manually put some red marks on the ground plane so that the corresponding point between top-view frame and side-view frame can be found. Note that in real world practical applications, any physical objects or features (such as road markings) can be used as equivalent markers. The motivation for using the red marks on the ground plane in this research is to ensure that corresponding points are obtained in a simple manner and to demonstrate the best possible results obtained using accurate homogrpahy matrices. After locating all corresponding points in each view, Random Sample Consensus (RANSAC) algorithm is leveraged to set an initial value of each homography matrix. In the last step we use Levenberg-Marquardt algorithms [24] to optimize the homography matrix.

2.3.1 Random Sample Consensus (RANSAC)

RANSAC is an iterative algorithm to estimate parameters of a mathematical model from a set of observed data that contains outliers, when outliers are to be accorded no influence on the values of the estimates. By setting a threshold of maximum iteration times and a threshold of

how much outliers can be removed, an initial homography matrix can be obtained that can map all inliers points of interest from top-view frames to side-view frames.

In the ground-plane image, let $\boldsymbol{q_a} = [u_a, v_a, 1]^T$ denotes the homogeneous coordinate of an interest point, where $a \in \{1,2,3 \dots A\}$ and A is the total number of interest points on the ground plane. Let $\boldsymbol{Q} = [\boldsymbol{q_1}, \boldsymbol{q_2}, \dots, \boldsymbol{q_A}]$ denotes a 3 × A matrix that each column of this matrix is the homogeneous coordinate of interest point in ground plane. In the side-view image that is captured by camera k, let $\boldsymbol{s_k^a} = [x_k^a, y_k^a, 1]^T$ denotes the homogeneous coordinate of the same interest point corresponding to $(u_a, v_a, 1)^T$, and $\boldsymbol{S_k} = [\boldsymbol{s_k^1}, \boldsymbol{s_k^2}, \dots, \boldsymbol{s_k^A}]$ denotes a 3 × A matrix that each column of this matrix is the homogeneous coordinate of interest point in side view k. As noted before, in this research the set of red markers shown in Fig.4 are used as corresponding interest points.

(a) Ground-plane frame from top-view camera

(a) Side-view frame from side-view camera k

Fig.4 Depiction of the process to infer the homography between camera and the ground plane

To calculate the initial $\boldsymbol{H}_k$, we apply the Direct Linear Transformation (DLT) algorithm in each iteration of RANSAC. We first randomly select 4 different interest points on the ground plane, and solve the equation in the form:

$$\begin{bmatrix} -x_k^r & -y_k^r & -1 & 0 & 0 & 0 & u_r x_k^r & u_r y_k^r & u_r \\ 0 & 0 & 0 & -x_k^r & -y_k^r & -1 & v_r x_k^r & v_r y_k^r & v_r \\ -x_k^s & -y_k^s & -1 & 0 & 0 & 0 & u_s x_k^s & u_s y_k^s & u_s \\ 0 & 0 & 0 & -x_k^s & -y_k^s & -1 & v_s x_k^s & v_s y_k^s & v_s \\ -x_k^i & -y_k^i & -1 & 0 & 0 & 0 & u_i x_k^i & u_i y_k^i & u_i \\ 0 & 0 & 0 & -x_k^i & -y_k^i & -1 & v_i x_k^i & v_i y_k^i & v_i \\ -x_k^j & -y_k^j & -1 & 0 & 0 & 0 & u_j x_k^j & u_j y_k^j & u_j \\ 0 & 0 & 0 & -x_k^j & -y_k^j & -1 & v_j x_k^j & v_j y_k^j & v_j \end{bmatrix} \begin{bmatrix} h_k^1 \\ h_k^2 \\ h_k^3 \\ h_k^4 \\ h_k^5 \\ h_k^6 \\ h_k^7 \\ h_k^8 \\ h_k^9 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.8)$$

In equation (2.8), $r, s, i, j \in \{1,2,3 \dots A\}$ and $r \neq s \neq i \neq j$. After $\boldsymbol{H}_k$ is calculated in the current iteration, we map all interest points in side view k to the ground plane and calculate the Euclidean distance between projected points and their corresponding points to evaluate the number of inliers interest points. Let $\widehat{\boldsymbol{q}}_{\boldsymbol{a}} = c_a[\hat{u}_a, \hat{v}_a, 1]^T$ be the projected points of $\boldsymbol{w}_{\boldsymbol{k}}^{\boldsymbol{a}}$ on the ground plane, where c_a is a non-zero constant and $a \in \{1,2,3 \dots A\}$. We then calculate the Euclidean distance between $\widehat{\boldsymbol{q}}_{\boldsymbol{a}}$ and $\boldsymbol{q}_{\boldsymbol{a}}$, and if the distance is smaller than the pre-set distance threshold, than $\boldsymbol{w}_{\boldsymbol{k}}^{\boldsymbol{a}}$ is classified as an inlier interest point. If the total number of inlier interest points is more than or equal to the inlier number threshold, we use the calculated $\boldsymbol{H}_k$ in current iteration as the initial homography matrix and terminate the iteration. If not, we repeat the entire process until the condition is satisfied.

2.3.2 Levenberg-Marquardt algorithms

The Levenberg-Marquardt algorithm is a non-convex optimization algorithm which provides a numerical solution to the problem of minimizing a non-linear function. It is fast and has stable convergence and is suitable for training small and medium sized problems.

After leveraging RANSAC algorithm, we have an initial homography matrix $\boldsymbol{H}_k$. Let $E(\boldsymbol{Q}, \boldsymbol{W}, \boldsymbol{H}_k)$ denotes the error function where,

$$E(\boldsymbol{Q}, \boldsymbol{W}, \boldsymbol{H}_k) = \frac{1}{2}\sum_{a=1}^{A} \|\boldsymbol{q_a} - \widehat{\boldsymbol{q}}_{\boldsymbol{a}}\| \tag{2.9}$$

Let $\boldsymbol{z_1} = [h_k^1, h_k^2, h_k^3, h_k^4, h_k^5, h_k^6, h_k^7, h_k^8, h_k^9]^T$ be the initial parameter vector whose elements are equal to each value in the initial homography matrix $\boldsymbol{H}_k$, and $\boldsymbol{z_n}$ denotes the updated parameter vector in the n^{th} iteration during the optimization process. The update rule of Levenberg-Marquardt algorithm can be presented as:

$$\boldsymbol{z_{n+1}} = \boldsymbol{z_n} - (\boldsymbol{J_n^T J_n} + \mu \boldsymbol{I})^{-1} \boldsymbol{J_n e_n} \tag{2.10}$$

In equation (2.10), $\boldsymbol{e_n}$ is the error vector which is calculated in each iteration of the update process where,

$$\boldsymbol{e_n} = \begin{bmatrix} e_{1,1} \\ e_{1,2} \\ e_{1,3} \\ e_{2,1} \\ e_{2,2} \\ e_{2,3} \\ e_{3,1} \\ e_{3,2} \\ e_{3,3} \\ . \\ . \\ . \\ e_{A,1} \\ e_{A,2} \\ e_{A,3} \end{bmatrix} = \begin{bmatrix} (u_1 - \hat{u}_1)^2 \\ (v_1 - \hat{v}_1)^2 \\ 0 \\ (u_2 - \hat{u}_2)^2 \\ (v_2 - \hat{v}_2)^2 \\ 0 \\ (u_3 - \hat{u}_3)^2 \\ (v_3 - \hat{v}_3)^2 \\ 0 \\ . \\ . \\ . \\ (u_A - \hat{u}_A)^2 \\ (v_A - \hat{v}_A)^2 \\ 0 \end{bmatrix} \tag{2.11}$$

and $\boldsymbol{J_n}$ is the Jacobian of $\boldsymbol{e_n}$ with respect to each parameter in the homography matrix. **The parameter** μ is a positive constant, called the combination coefficient.

2.4 Multi-View ROI Inference

From the estimated homography matrices, each $\boldsymbol{C}_{i,j}^{t}$ in every MER is projected to each side view. We denote the projected pixel of $\boldsymbol{C}_{i,j}^{t}$ in side view k as $\boldsymbol{P}_{i,j,k}^{t}$. A set of bounding boxes are then generated according to the projected pixels, where each pixel associates with L multi-scale-multi-aspect-ratio bounding boxes. The bottom edge of each bounding box is centered at the corresponding projected pixel [15]. We denote the l^{th} bounding box whose bottom edge is centered at the $\boldsymbol{P}_{i,j,k}^{t}$ as $\boldsymbol{A}_{i,j,k,l}^{t}$, where $l \in \{1,2,\dots,L\}$ and L is the total number of bounding boxes associated with $\boldsymbol{P}_{i,j,k}^{t}$. In this work, bounding boxes with 3 different scales and 3 different aspect ratios are used, and hence $L = 9$ for each projected pixel. AlexNet, a pre-trained deep CNN $\mathcal{F}\left(\boldsymbol{A}_{i,j,k,l}^{t}, \boldsymbol{I}_{k}^{t}\right)$ is fine-tuned by transfer learning to assign the probability to each bounding box. Since the AlexNet is a pre-trained deep CNN, in this research we only need to train the last few MLP structure classification layer from scratch and the convolutional layers of AlexNet is stable during the training process. The reason we leverage a pre-trained CNN in this research is that we only need to extract feature from the image patch that is cropped by each bounding box, and then classify whether this image patch is a vehicle. Besides the training data for the CNN is limited. Hence a pre-trained CNN is preferred in this research since the data needed to only train the classification layer of a pre-trained CNN is much less than the data needed to train a CNN from scratch. The maximum probability of the bounding box being vehicle is assigned to the MER $\boldsymbol{R}_{i}^{t}$ on the ground plane as:

$$\Pr(\boldsymbol{R}_{i}^{t}|G^{t}) = \max_{j,k,l} \mathcal{F}\left(\boldsymbol{A}_{i,j,k,l}^{t}, \boldsymbol{I}_{k}^{t}\right) \tag{2.12}$$

The probability assignment process is illustrated in Fig.2, where $\boldsymbol{A}_{2,1,1,2}^{t}$ is assigned the maximum probability and the false positive MER $\boldsymbol{R}_{1}^{t}$ is eliminated by multi-view ROI inference. The state $S(\boldsymbol{R}_{i}^{t}|G^{t})$ of the MER $\boldsymbol{R}_{i}^{t}$ is estimated using probability thresholding as:

$$S(\boldsymbol{R}_{i}^{t}|G^{t}) = \begin{cases} 0, \text{if } \Pr(\boldsymbol{R}_{i}^{t}|G^{t}) \leq a \\ 1, \text{otherwise} \end{cases} \tag{2.13}$$

where $a \in [0,1]$ is the probability threshold. The threshold a is determined such that the prediction results yield the highest performance in validation set. The proposed system recalls $\boldsymbol{R}_{i}^{t}$ as the vehicle when $S(\boldsymbol{R}_{i}^{t}|G^{t}) = 1$ and eliminates $\boldsymbol{R}_{i}^{t}$ when $S(\boldsymbol{R}_{i}^{t}|G^{t}) = 0$.

III. Experimental Results

In this section, we present experimental results of the proposed automatic multi-camera vehicle detection system. The experiments are conducted on real-traffic image data that is captured from a roadway in Richardson, TX, USA.

3.1 Data Preparation

The synchronized image data is captured from 4 cameras as shown in Fig.3. The captured frames are sampled such that the number of frames with vehicles are equal to those without vehicles. The remaining 9960×4 frames are split in the proportion of 3:1:1 to correspond respectively to training, validation and test sets. For MVRPN training, the synchronized dimension-reduced frames of 3 side cameras are used as inputs. The target top-view frames are labeled as pixel-wise binary masks, where the positives indicate the vehicle and the negatives indicate the background on the ground plane. Note that the training samples are input into the MVRPN randomly rather than chronologically. For CNN training, the ground-truth bounding boxes are labeled at 3 side views, and image patches are then extracted by applying Edge Boxes [25]. The extracted image patches whose Intersection over Union (IoU) with a ground-truth bounding box greater than 0.7 are treated as positives; IoU less than 0.3 are treated as negatives; and the rest are ignored. The ratio of the positive samples to the negative samples is set to 1:2.

3.2 Modeling Training Configuration

All the experiments are performed using a desktop with Intel (R) Quad-Core (TM) i5-7400 CPU@3.0GHz Processor, 8GB RAM, and NVIDIA GeForce GTX 1050Ti 4GB GPU.

3.2.1 Multi-view region proposal network

The MVRPN is trained by minimizing the loss function in Eq. 1. The synchronized side-view frames are RGB images. The 1500×1 MVRPN input vector is obtained by retaining the first 500 principal components for each of the 3 side views. Ground-truth occupancy vectors are obtained by subsampling 300×600 ground-plane binary mask into 15×30 grid of cells. During training process, RMSProp [26] with 128 batch size, 0.15 initial learning rate, $\eta^+ = 1.2$, and $\eta^- = 0.5$ is applied. During the optimization process of RMSProp algorithm, if the sign of the last two gradients of the loss function are the same, which means that the loss function still has not achieved the local minimum, then we multiplicatively increase the learning rate by a factor η^+. If the sign of the last two gradients of loss function are different, then we multiplicatively decrease the learning rate by factor η^-.

3.2.2 Transfer learning prediction

The fine tuning of the pre-trained AlexNet is implemented on MATLAB R2017b with AlexNet support package. During the training process, stochastic gradient descent (SGD) [27] with 128 batch size, 0.9 momentum, 10^{-4} initial learning rate, and 10^{-4} L_2 regularization is applied.

3.3 Comparative Evaluation

Table 2. Numeric Evaluation Results

Camera deployments	AP	MODP
$\boldsymbol{C_{1,2,3}}$	**0.7849**	**0.7089**
$C_{1,2}$	0.6087	0.6526
$C_{1,3}$	0.5989	0.6554
$C_{2,3}$	0.6761	0.6722
C_1	0.4401	0.6175
C_2	0.5124	0.6287
C_3	0.4673	0.6208

*Note: $C_{\alpha,\beta,\gamma}$ represents utilization of side camera α, β, and γ.

We evaluate the multi-camera vehicle detection system on 1992 top-view test images. To our best knowledge, there is no published dataset about multi-camera vehicle detection. The feature extracted in the existing multi-camera pedestrian detection algorithm is not applicable in this paper [14], [18], [19], [20]. Hence, we benchmark the performance of the multi-camera vehicle detection system by deploying different camera combinations. For the fixed IoU, the system is evaluated by Average Precision (AP) [28] and Multiple Object Detection Precision (MODP) [29]. The detected bounding boxes are considered as true positives when the IoUs exceed 0.55. The precision-recall curve is shown in Fig.5. The result shows that using all 3 cameras provides a significant improvement in the performance compared to using only a single camera. For comparison, the performance of the system when only 2 cameras are used is also given. For all 3 combinations of 2-camera systems the performance is better than using a single camera but inferior to that of using 3 cameras. For the varying IoUs, Multiple Object Detection Accuracy (MODA) curve [29] is shown in Fig.6. As with the earlier results, the performance increases as the number of cameras increases from 1 to 3. The evaluation results of AP and

MODP are shown in Table 2, where the camera deployment $C_{1,2,3}$ achieves the best performance (0.7849 AP and 0.7089 MODP) among all variations. The utilizations of 2 side-view cameras achieve better performances than single camera deployments. Such numeric evaluation results indicate that the performance of the multi-camera vehicle detection system increases when more side-view cameras are deployed.

Table 3. Definition of Evaluation Metrics

	Description
Precision	$\frac{TP}{TP + FP}$
Recall	$\frac{TP}{TP + FN}$
Average Precision	$\frac{1}{11} \sum_{r \in \{0.0, 0.1, \ldots, 1.0\}} Precision_r$
MODA	$1 - \frac{FP + FN}{TP + FN}$
MODP	$\frac{Mapped\ Overlap\ Ratio}{N_{mapped}}$, where $Mapped\ overlap\ ratio = \sum_{i=1}^{N_{mapped}} \frac{G_i \cap D_i}{G_i \cup D_i}$

*Note: TP means true positive prediction, FP means false positive prediction and FN means false negative prediction.

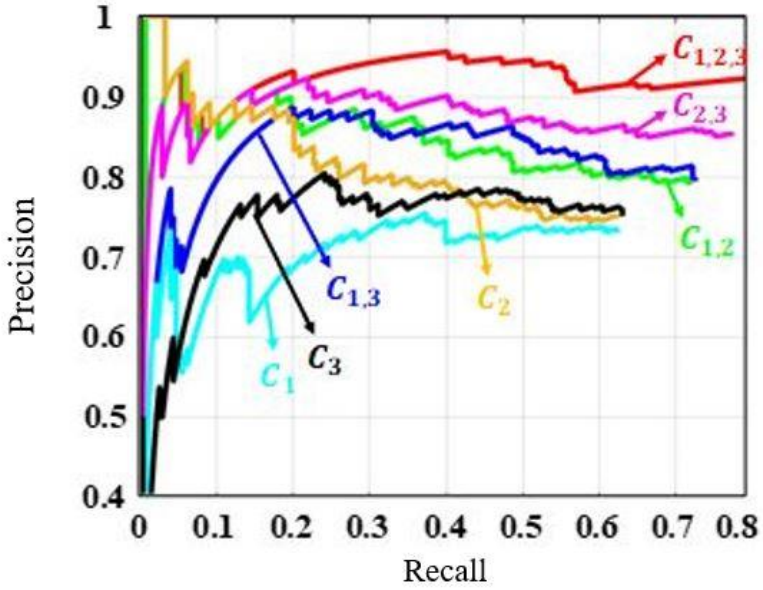


Fig. 5. Precision-Recall curve.

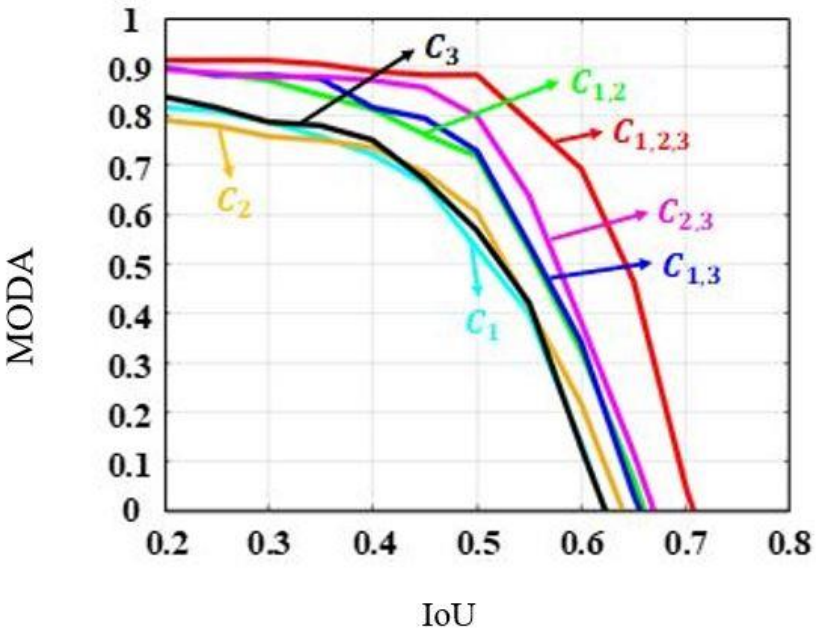


Fig. 6. MODA curve.

3.4. Visualization Results

Examples of vehicles detected on the ground plane using $C_{1,2,3}$ are shown in Fig.6 and Fig.7. The number on each bounding box is the probability that indicates whether the object enclose by that bounding box is a vehicle. According to the detection results, it is clear that the system can detect vehicles with varying sizes, e.g. the white sedan vs. the yellow SUV in Fig.6(a). The partially-observed black SUV with smaller size than regular vehicle is also detected in Fig.6(a). However, the detected bounding box of the yellow vehicle at Fig.6(a) is not of optimal shape and size, and the partially-observed vehicle at right boundary of Fig.7(a) is not detected. Note that in Fig.7(b), there are two vehicles are not detected. The reason is that those two vehicles are not captured by the top-view camera, which means those two vehicles are not in the ground plane field, and the proposed system can only detect vehicles that are also in the ground plane field. The proposed system can cover a wider area if we enlarge the field of view of the top camera in the training process.

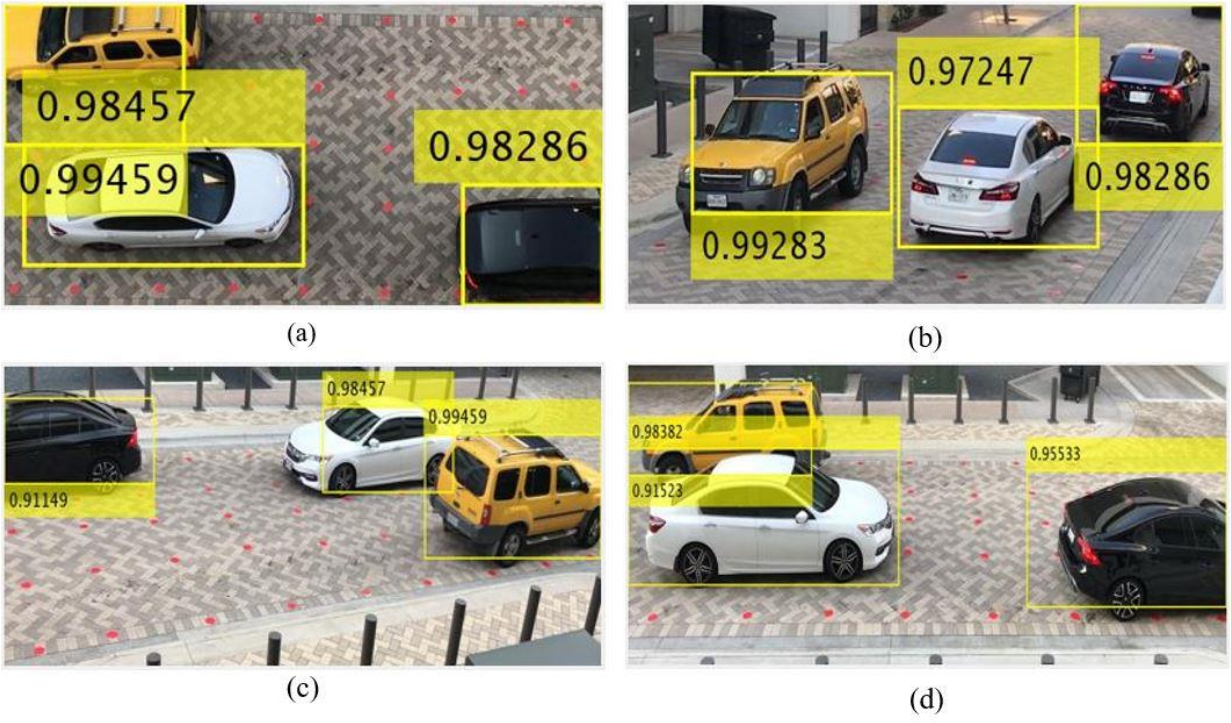


Fig. 6. Synchronized detection results (1). (a) is the detection result on the ground plane. (b), (c) and (d) is the detection result in side view 1, side view 2 and side view 3 separately.

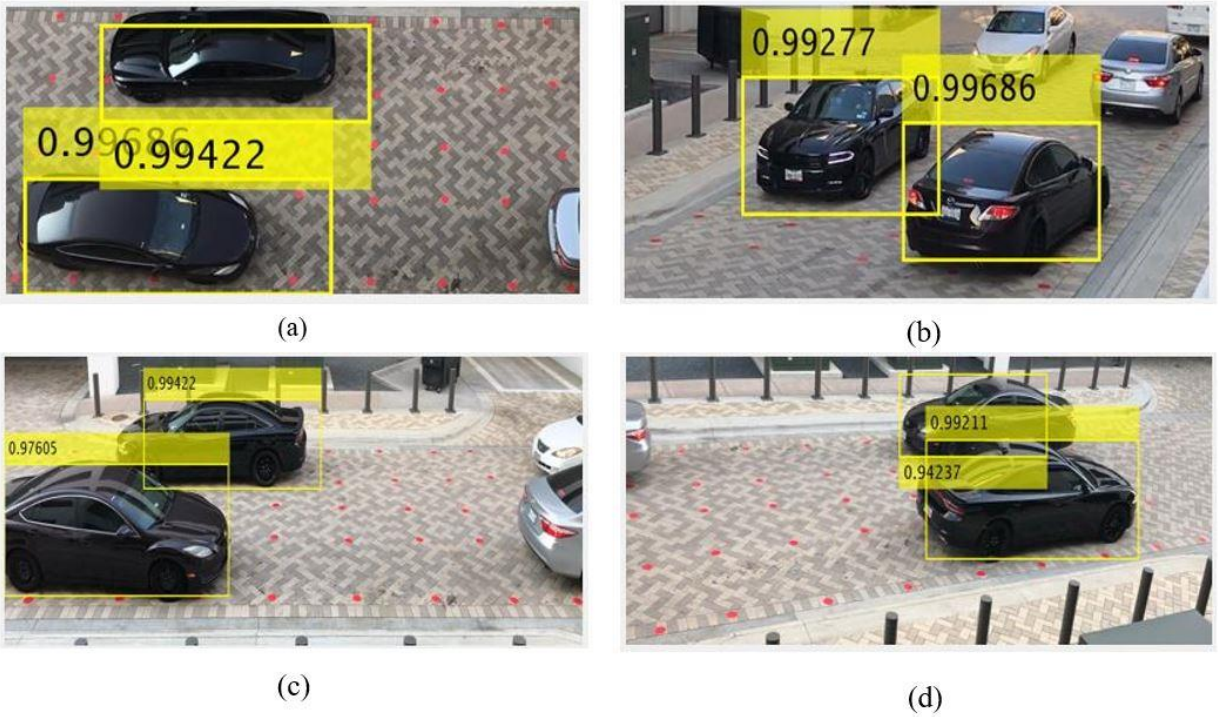


Fig. 7. Synchronized detection results (2). (a) is the detection result on the ground plane. (b), (c) and (d) is the detection result in side view 1, side view 2 and side view 3 separately.

IV. Conclusion

In this thesis, a multi-camera vehicle detection system with a MVRPN/CNN pipeline is presented. The Multi-Layer Perceptron structure MVRPN is constructed to produce the candidate location of vehicle on the ground plane which. The output of MVRPN may contain some false positive predictions. The pre-trained fine-tuned CNN is utilized to remove those false positive prediction by projecting all cells may occupied by vehicle on the ground plane back to each side view and infer the probability whether the cell is occupied by a vehicle or not. Moreover, since we use block of cells rather than a single block on the ground plane to represent location of vehicles, the proposed system can be utilized to detect vehicle with large variations in size and shape. The experiments result shows that our approach achieves a better performance if we utilize more cameras to construct the camera network.

The proposed system is based on a hypothesis that vehicle occluded in some views may not be occluded in other views. However, sometimes a vehicle may be occluded in all views so that the system cannot detect the totally occluded vehicle in some frames. Hence in future investigations, a vehicle detection system which can utilize temporal video frames will be developed to address vehicle tracking-related challenges, so even in some frames that some vehicles are occluded in all views, the system can still predict the location of those totally occluded vehicles by utilizing information in temporal neighbor frames. In addition, a multi-view bounding-box regression will be embedded into the pipeline to optimize the bounding-box predictions. Future work will also have to consider optimal strategies to determine the locations for the various cameras as well as the cost-benefit analysis of increasing the number of cameras. The robustness of the proposed approach when it is applied in slightly different contexts to where it was trained should also be investigated.

BIBLIOGRAPHY

[1] L. E. Y. Mimbela, and L. A. Klein, “Summary of vehicle detection and surveillance technologies used in intelligent transportation systems,” Federal Highway Administration, Tech. Rep., 2000.

[2] Z. Sun, G. Bebis, and R. Miller, “On-road vehicle detection: A review,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 5, pp. 694-711, 2006.

[3] Y. Wang, Y. Huang, W. Zheng, Z. Zhou, D. Liu, and M. Lu, “Combining convolutional neural network and self-adaptive algorithm to defeat synthetic multi-digit text-based CAPTCHA,” In Proceedings of the IEEE International Conference on Industrial Technology, Mar. 2017, pp. 980-985.

[4] Y. Wang, Z. Zhou, S. Jin, D. Liu, and M. Lu, “Comparisons and selections of features and classifiers for short text classification,” In IOP Conference Series: Materials Science and Engineering, vol. 261, no. 1, pp. 012018, 2017.

[5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, “SSD: Single Shot Multibox Detector,” In Proceedings of the European Conference on Computer Vision, Oct. 2016, pp. 21-37.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 779-788.

[7] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” In Advances in Neural Information Processing Systems, 2015, pp. 91-99.

[8] H. Possegger, T. Mauthner, P. M. Roth, and H. Bischof, “Occlusion geodesics for online multi-object tracking,” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1306-1313.

[9] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan, “Vision-Based Occlusion Handling and Vehicle Classification for Traffic Surveillance Systems,” IEEE Intelligent Transportation Systems Magazine, vol. 10, no. 2, pp. 80-92, 2018.

[10] B. Tian, Y. Li, B. Li, and D. Wen, “Rear-view vehicle detection and tracking by combining multiple parts for complex urban surveillance,” IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 2, pp. 597-606, 2014.

[11] B. Tian, M. Tang, and F. Y. Wang, “Vehicle detection grammars with partial occlusion handling for traffic surveillance,” Transportation Research Part C: Emerging Technologies, vol. 56, pp. 80-93, 2015.

[12] C. Wang, Y. Fang, H. Zhao, C. Guo, S. Mita, and H. Zha, “Probabilistic inference for occluded and multiview on-road vehicle detection,” IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 1, pp. 215-229, 2016.

[13] M. Hofmann, D. Wolf, and G. Rigoll, “Hypergraphs for joint multi-view reconstruction and multi-object tracking,” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3650-3657.

[14] X. Wang, “Intelligent multi-camera video surveillance: A review,” Pattern recognition

letters, vol. 34, no. 1, pp. 3-19, 2013.

[15] K. Kim, and L. S. Davis, “Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering,” In Proceedings of European Conference on Computer Vision, May. 2006, pp. 98-109.

[16] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua, “Multicamera people tracking with a probabilistic occupancy map,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 2, pp. 267-282, 2008.

[17] A. C. Sankaranarayanan, A. Veeraraghavan, and R. Chellappa, “Object detection, tracking and recognition for multiple smart cameras,” Proceedings of the IEEE, vol. 96, no. 10, pp. 1606-1624, 2008.

[18] P. Peng, Y. Tian, Y. Wang, J. Li, and T. Huang, “Robust multiple cameras pedestrian detection with multi-view Bayesian network,” Pattern Recognition, vol. 48, no. 5, pp. 1760-1772, 2015.

[19] P. Baqué, F. Fleuret, and P. Fua, “Deep occlusion reasoning for multi-camera multi-target detection,” In Proceedings of the IEEE International Conference on Computer Vision, Oct. 2017, vol. 2.

[20] T. Chavdarova, “Deep multi-camera people detection,” In Proceedings of the IEEE International Conference on Machine Learning and Applications, Dec. 2017, pp. 848-853.

[21] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, “Model-based compressive sensing,” IEEE Transactions on Information Theory, vol. 56, no. 4, pp. 1982-2001, 2010.

[22] I. Jolliffe, “Principal component analysis,” In International Encyclopedia of Statistical Science, pp. 1094-1096, 2011.

[23] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic

segmentation," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431-3440.

[24] R. Hartley, and A. Zisserman, Multiple view geometry in computer vision, Cambridge university press, 2003.

[25] C. L. Zitnick, and P. Dollár, "Edge boxes: Locating object proposals from edges," In Proceedings of the European Conference on Computer Vision, Sep. 2014, pp. 391-405.

[26] T. Tieleman, and G. Hinton, "Lecture 6.5-RMSProp: Divide the gradient by a running average of its recent magnitude," COURSERA: Neural networks for machine learning, vol. 4, no. 2, pp. 26-31, 2012.

[27] L. Bottou, "Large-scale machine learning with stochastic gradient descent," In Proceedings of COMPSTAT, 2010, pp. 177-186.

[28] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," International Journal of Computer Vision, vol. 88, no. 2, pp. 303-338, 2010.

[29] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V.Korzhova, J. Zhang, "Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 2, pp. 319-336, 2009.